

CLAIMS

What is claimed is:

1. A method comprising:
loading a table having a set of L data elements;
determining whether said table fits into a single register;
performing a data lookup into said table with a packed data shuffle operation if said determination indicates that said table does fit into a single register; and
dividing said table into a plurality of sections if said table does not fit into a single register, each of said sections sized to fit into a single register, and executing a plurality of packed data shuffle operations on said plurality of sections to look up data in said table.
2. The method of claim 1 further comprising loading a lookup mask for each packed data shuffle operation, said lookup mask to indicate which data elements are to be extracted from said table.
3. The method of claim 2 wherein said lookup mask is comprised of L shuffle masks, each shuffle mask corresponding to a unique data element position.
4. The method of claim 3 wherein each shuffle mask is comprised of:
a flush to zero field, said flush to zero field to indicate whether a data element position associated with this shuffle mask is to be filled with a zero value;
a selection field, said selection field to indicate which table data element to shuffle data from; and
a source select field, said source select field to indicate which of said plurality of table sections to shuffle data from for this shuffle mask.

5. The method of claim 2 further comprising merging shuffle results from said plurality of packed data shuffle operations into a single register.
6. The method of claim 3 wherein each packed shuffle operation comprises:
for each shuffle mask, shuffling data from a data element designated by said shuffle mask to an associated resultant data element position if its flush to zero field is not set and placing a zero into said associated resultant data element position if its flush to zero field is not set.
7. The method of claim 6 wherein a capacity of a single register is 128 bits.
8. The method of claim 7 wherein each data element is a byte wide and each shuffle mask is a byte wide.
9. The method of claim 8 wherein said lookup mask is 128 bits long and L is less than seventeen.
10. A method for table lookup comprising:
loading data for a first M-bits wide portion and data for a second M-bits wide portion of a table;
loading an M-bits wide mask, said mask comprised of N control elements, each control element corresponding to a unique data element position;
shuffling said first M-bits wide portion in accordance to said M-bits wide mask to generate a first shuffled result;
shuffling said second M-bits wide portion in accordance to said M-bits wide mask to generate a second shuffled result;
merging selected data elements from said first and second shuffled results to obtain an M-bits wide table lookup resultant.

11. The method of claim 10 wherein said table and said portions of said table are comprised of packed data elements.
12. The method of claim 11 wherein said first M-bits wide portion, said second M-bits wide portion, and said M-bits wide table lookup resultant are each comprised of N packed elements.
13. The method of claim 11 wherein M is 128 and N is 16.
14. The method of claim 12 wherein each control element is comprised of:
 - a flush to zero field, said flush to zero field to indicate whether a data element position associated with this control element is to be filled with a zero value;
 - a selection field, said selection field to indicate which table data element to shuffle data from; and
 - a source select field, said source select field to indicate which of said plurality of table sections to shuffle data from for this control element.
15. The method of claim 10 further comprising generating a table select mask from M-bits wide mask, said table select mask to indicate which table section each resultant data element position should receive data from.
16. The method of claim 15 further comprising:
 - applying said table select mask to said first shuffled result, wherein a first shuffled data element is selected from said first shuffled result; and
 - applying said table select mask to said second shuffled result, wherein a second shuffled data element is selected from said second shuffled result.
17. The method of claim 16 wherein said merging selected data elements comprises merging data from said first shuffled data element and said second shuffled data element

into said M-bits wide table lookup resultant, data from said first and data from said second shuffled data elements are to each occupy a separate data element position.

18. The method of claim 10 further comprising determining whether said table for said table lookup can fit into a single register, where if true, performing said table lookup with a shuffle operation on said table with said M-bits wide mask instead of performing lookups on multiple portions of said table.

19. The method of claim 18 wherein said single register is a 128 bit wide single instruction multiple data register, M less than 129, and said table is less than 129 bits wide.

20. An article comprising a machine readable medium that stores a program, said program being executable by a machine to perform a method comprising:

determining whether a table having a set of L data elements fits into a single register;

performing a data lookup into said table with a packed data shuffle operation if said determination indicates that said table does fit into a single register; and

dividing said table into a plurality of sections if said table does not fit into a single register, each of said sections sized to fit into a single register, and executing a plurality of packed data shuffle operations on said plurality of sections to look up data in said table.

21. The article of claim 20 wherein said method further comprises loading a lookup mask for each packed data shuffle operation, said lookup mask to indicate which data elements are to be extracted from said table.

22. The article of claim 21 wherein said lookup mask is comprised of L shuffle

masks, each shuffle mask corresponding to a unique data element position.

23. The article of claim 22 wherein each shuffle mask is comprised of:

a flush to zero field, said flush to zero field to indicate whether a data element position associated with this shuffle mask is to be filled with a zero value;

a selection field, said selection field to indicate which table data element to shuffle data from; and

a source select field, said source select field to indicate which of said plurality of table sections to shuffle data from for this shuffle mask.

24. The article of claim 20 wherein said program further comprises merging shuffle results from said plurality of packed data shuffle operations into a single instruction multiple data register.

25. The article of claim 23 wherein each packed shuffle operation comprises:

for each shuffle mask, shuffling data from a data element designated by said shuffle mask to an associated resultant data element position if its flush to zero field is not set and placing a zero into said associated resultant data element position if its flush to zero field is not set.

26. The article of claim 25 wherein each data element is a byte wide and each shuffle mask is a byte wide.

27. The article of claim 26 wherein said single register has a capacity of 128 bits and L is less than seventeen.

28. An apparatus comprising:

an execution unit to execute a sequence of instructions, said instructions to perform a table lookup operation, said instructions to cause said execution to:

determine whether a table having a set of data elements fits into a single register;

perform a data lookup into said table with a packed data shuffle operation if said determination indicates that said table does fit into a single register; and

divide said table into a plurality of sections if said table does not fit into a single register, each of said sections sized to fit into a single register, and execute a plurality of packed data shuffle operations on said plurality of sections to look up data in said table.

29. The apparatus of claim 28 wherein said instructions are to further cause said execution unit to load a lookup mask for each packed data shuffle operation, said lookup mask to indicate which data elements are to be extracted from said table.

30. The apparatus of claim 29 wherein said lookup mask is comprised of a plurality of shuffle masks, each shuffle mask corresponding to a unique data element position.

31. The apparatus of claim 30 wherein each shuffle mask is comprised of:

a flush to zero field, said flush to zero field to indicate whether a data element position associated with this shuffle mask is to be filled with a zero value;

a selection field, said selection field to indicate which table data element to shuffle data from; and

a source select field, said source select field to indicate which of said plurality of table sections to shuffle data from for this shuffle mask.

32. The apparatus of claim 31 wherein said execution is to comprises merging shuffle results from said plurality of packed data shuffle operations and to store said merged shuffle results into a single instruction multiple data register.

33. The apparatus of claim 31 wherein each packed shuffle operation comprises:
- for each shuffle mask, shuffling data from a data element designated by said shuffle mask to an associated resultant data element position if its flush to zero field is not set and placing a zero into said associated resultant data element position if its flush to zero field is not set.
34. The apparatus of claim 33 wherein each data element is a byte wide and each shuffle mask is a byte wide.
35. A system comprising:
- a memory to store data and instructions;
 - a processor coupled to said memory on a bus, said processor operable to perform instructions for a table lookup algorithm, said processor comprising:
 - a bus unit to receive a sequence of instructions from said memory;
 - an execution unit coupled to said bus unit, said execution unit to execute said sequence, said sequence to cause said execution unit to:
 - determine whether a table having a set of data elements fits into a single register;
 - perform a data lookup into said table with a packed data shuffle operation if said determination indicates that said table does fit into a single register; and
 - divide said table into a plurality of sections if said table does not fit into a single register, each of said sections sized to fit into a single register, and execute a plurality of packed data shuffle operations on said plurality of sections to look up data in said table.

36. The system of claim 35 wherein said instructions are to further cause said execution unit to load a lookup mask for each packed data shuffle operation, said lookup mask to indicate which data elements are to be extracted from said table.

37. The system of claim 36 wherein said lookup mask is comprised of a plurality of shuffle masks, each shuffle mask corresponding to a unique data element position, and wherein each shuffle mask is comprised of:

a flush to zero field, said flush to zero field to indicate whether a data element position associated with this shuffle mask is to be filled with a zero value;

a selection field, said selection field to indicate which table data element to shuffle data from; and

a source select field, said source select field to indicate which of said plurality of table sections to shuffle data from for this shuffle mask.

38. The system of claim 37 wherein each packed shuffle operation comprises:

for each shuffle mask, shuffling data from a data element designated by said shuffle mask to an associated resultant data element position if its flush to zero field is not set and placing a zero into said associated resultant data element position if its flush to zero field is not set.

39. The system of claim 38 wherein said execution is to comprises merging shuffle results from said plurality of packed data shuffle operations and to store said merged shuffle results into a single instruction multiple data register.

40. The system of claim 39 wherein each data element is a byte wide and each shuffle mask is a byte wide.